Victorian 6502 User Group Newsletter

KAOS

For People Who Have Got Smart

OSI SYM KIM AIM UK101 RABBLE 65

Registered by Australia Post
Publication No. VBG4212

ADDRESS ALL CORRESPONDENCE TO 10 FORBES ST., ESSENDON, VIC. 3040

Vol.4 No.6 March 1984

Our first task this month is to up-date our heading. Rod Drysdale is no longer able to help members who are interested in Amateur Radio and Clive Harman has offered to take over. Clive can be contacted at P . His call sign is VK3BUS. Rod has been looking after the interests of amateurs since the club started and we thank him sincerely for the work he has done.

Also, our Forth contact, David Wilson, is enjoying an extended overseas trip and until he returns and is in a position to help members again, would you please contact KAOS and we will put you in touch with a member who can assist you.

As you see by the heading on his article, David Dodds has finished his series on machine language. David originally offered to do a 'few' articles on machine language but things seemed to get out of hand and after nearly two years David has finally decided to call a halt. We would like to thank David for the time and effort he has put into the series, especially for the times he was busy with other matters, (including a new son) and always managed to have the article ready when we needed it. We will be looking at YOU for articles to fill the gap the end of his series will leave.

John Whitehead has compiled a list of the first items that have arrived in response to his request for information on items of hardware and software that members have for sale. You will find the list in this issue, if your special project is not there, contact John at and he will include it in the next list.

The next meeting will be at 2pm on Sunday 25th March 1984 at the Essendon Primary School on the corner of Raleigh and Nicholson Streets, Essendon. Those members that help with the school children please note that the school will be open at 9.30am to allow time to set up your machines, all other members are welcome at 1pm.

The closing date for articles for the April newsletter is the 13th April.

INDEX

FORTH SCREENS

by Ron Cork

Now that I have an RGB colour monitor, I have been playing around with my various disks, setting them up with different colours for different routines and applications. Listed are a few FORTH screens that will fill full video-screens with a colour of your choice, fill only the top or bottom half of the video, or allow you to place a horizontal 'STRIP' or 'STRIPE' of any colour you wish to any location on the video-screen.

The STRIP is a line of colour in only the normal window area of the monitor, while the STRIPE will do the full video-screen width, including the borders. The listings should be self-explanatory with one possible exception. In screen 14, there is a non-standard word 'STOP' defined by Ray Gardiner, which simply does a screen-clear and sets the cursor to the normal lower left-hand corner.

Screen 14 defines a word 'BS' which uses STOP to clear the video-screen, paint the screen blue, then sets-up the 'input' line, (that's the line where all your typing goes), as an inverse colour, to highlight your input.

Because a lot of commands are used very frequently during a FORTH session, I decided to shorten these commands to give me two-key entry style that speeds up my fooling around. There is plenty of scope here for more, it all depends on your own needs. You do need to be careful in your choice of WORDS to use. Check the vocabulary(s) first before you go re-defining existing WORDS.

You will notice 'SCAN' as one of these re-defined WORDS. This is another of Ray's. If you haven't got a listing or implimentation of these WORDS on your system, then I suggest you see Ray.

I must thank Stuart Thomas for giving me the original push in the right direction. A young man who should go far in programming , regardless of languages. He is presently setting up a Forth look-a-like system on his 8K Superboard, written in BASIC. Hard to believe, but it works. It is not a 'full-blown' Forth, but it will give the un-initiated a good idea of what Forth is all about and may provide the push needed to start other users into Forth.

```
SCR # 10
( Screen Wash Colours                    )

DECIMAL
; WHT 0 0 ;      : BLA 1 1 ;      : SKY 2 2 ;
: PUR 4 4 ;      : BLU 6 6 ;      : YEL 8 8 ;  : NUL 1 1 ;
: GRE 10 10 ;    : RED 12 12 ;    : INV 1 + SWAP ;
HEX
: COLOUR-ON   5 DE00 C! ;
: COLOUR-OFF  1 DE00 C! ;
: WASH
   E000 800 2OVER DROP FILL ( fill screen with one   )
       COLOUR-ON ;          ( colour and turn on     )
DECIMAL

;S

```

```
SCR # 11
0  ( Screen colours cont....                      )
1
2  HEX
3
4  : HIWASH E000 440 2OVER DROP     ( top half of screen      )
5         FILL 2DROP ;
6  : LOWASH E440 400 2OVER DROP     ( bottom half of screen   )
7         FILL 2DROP ;
8
9  DECIMAL
10
11 ( load screen 14 after this one for the rest of the screen )
12 ( colour definitions & screen 17 for short-form commands )
13 14 LOAD 17 LOAD
14 ;S
15
```

```
SCR # 14
0  ( Strip, Stripe and BS screen colours     )
1  HEX
2
3  : STRIP             ( fill one line in window area only )
4      39 SWAP 40 * E001 + SWAP
5      2OVER DROP FILL 2DROP ;
6  : STRIPE            ( fill one line for full screen width )
7      40 * SWAP 40 * E000 + SWAP
8      2OVER DROP FILL COLOUR-ON ;
9  DECIMAL
10 : BS                ( BS = Bootscreen colours )
11     BLU WASH RED INV 29 STRIP STOP ;
12 DISK HO             ( 29 is the 'data' input line )
13 ;S                  ( i.e... the 'cursor' line .   )
14                     ( this line is 'colour-inverted' )
15   ( NB...'STOP' is a 'Ray Gardiner' word.  May not apply. )
```

```
SCR # 17
0  (Short-form commands -- List,Edit,Flush,Vlist,etc....    )
1
2   : L. LIST ;
3   : E. EDIT ;
4   : S. SCAN ;
5   : F. FLUSH ;
6   : H. HEX ;
7   : DL DECIMAL ;
8   : V. VLIST ;
9   ;S
10
11 ( Be careful of any additions.  Check the vocabularies )
12 ( before anymore Short-form commands are added. )
13
14
15
```

Superboard

March, 1984.

NEWSLETTER OF THE OHIO SUPERBOARD USER GROUP, 146 YORK STREET, NUNDAH, 4012.

SOFTWARE REVIEW - Night Rider

Night Rider is an arcade game which is a Basic / M/C mix. The object of the game is to stay on a winding road, driving as fast as you can. You have three keys for control. B and N are for steering, and the left shift key is the accelerator. There are no brakes, but you decelerate quickly when you release the shift key. At the start of the game, you have the choice of seeing the car, which is a rather weak graphic considering that it does not have to move on the screen. There are digital displays of speed, time remaining, and distance travelled. The night road consists of white signposts, and there are dots to designate the road centre, and depict forward movement.

At low speeds, the steering is rather jerky, but at high speed, it's quite different. Super smooth 256 character resolution is achieved by using characters 136 through 143. The resolution change is fairly subtle at all but the very low speeds, making the game more realistic. Naturally, a vehicle won't turn as sharply at a higher speed. One of the weakest spots in the game occurs when you run off the road. An extremely poor representation of a crash is given, and you can continue on again as if nothing had happened. That certainly would not be the case if a vehicle travelling at speed hit a real signpost! Fortunately, this part of the program is in Basic, so it can be easily modified if you so desire. (As can the shape of the car)

When the timer counts down to zero, the game is over, and the program tells you the distance covered in miles. For your next game, you can have a re-run over the same road, or choose another.

When friends visit, and ask what your computer can do, they don't really want to see your wordprocessor, address lists, or financial management and other educational programs. This is an excellent game for a demo. Any driver can relate to it immediately, and soon master the controls and skills needed to attain a good score.

Night Rider will run on a C1 or C4, and should be fairly simple to modify to suit any format/keyboard combination. The best way to get the program is as part of the Victory Package, reviewed in March 83 KAOS, and on special offer to SUPERBOARD readers. (See July 83 KAOS)

The OSUG library has one for participating members at the usual postage rates.

LOAN REPAYMENTS

Ever wanted to figure out the monthly payments on a $30,000 loan? This program will do it for you. With a little modification it can do more. It assumes compounding interest (the usual type).

```
10 INPUT "Principal";P
20 INPUT "Interest";I
30 INPUT "Number of payments per annum";N
40 INPUT "Number of years of repayments";Y
50 R=((I/100)*P/N)/(1-1/((I/100)/N+1)↑(N*Y))
60 PRINT INT(R*100)/100;"Dollars"; INT(R*N*Y);"Total"
70 PRINT:PRINT:END
```

Oh well, perhaps you didn't need the loan after all! The total is a shock.

— SUPERBOARD —

THE NOS BASICODE

The NOS Basicode project was developed in The Netherlands as a result of a Radio program named Hobbyscoop. Initially, the station broadcast Basic programs directly from the cassette output of the four most popular makes of microcomputer at the time, ie the TRS-80, PET, APPLE and SORCERER. However this meant that a lot of on-air time was wasted by retransmitting virtually the same program in the different formats. As the number of makes of microcomputer increased, this process took up more and more time, as all the manufacturers used different frequencies, baud rates, and formats.

The purpose of NOS BASICODE is to standardise the frequencies, baud rates, and format stored on cassette, so that BASIC programs so stored can be loaded into any type of microcomputer. It then remains to sort out the small differences in the BASIC dialect that occur between various machines. What follows is a translation program that allows the OSI C1P to both save and load BASICODE programs. A Baud rate modification is also required.

```
ADDRESS 0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F

  1E00  A5 85 A4 86 85 E0 84 E1 A9 00 A0 1E 85 85 84 86
  1E10  85 83 84 84 A9 22 A0 1E 85 0B 84 0C A2 FF 9A 4C
  1E20  00 00 A9 52 A0 1E A2 00 8E 03 02 8E 05 02 20 C3
  1E30  A8 86 5F 20 EB FF C9 52 F0 0B C9 4C F0 1D C9 53
  1E40  D0 E0 4C 58 1F A5 E0 A4 E1 85 85 84 86 85 83 84
  1E50  84 60 20 53 2F 4C 2F 52 20 3F 00 20 CD 1E 20 F4  Cegmon 1A
  1E60  FF A9 00 85 E6 A5 85 A4 86 85 E2 84 E3 A9 FD 8D
  1E70  00 DF A9 10 2C 00 DF F0 29 20 EB FF A4 E6 D0 06  C4 Keybd
  1E80  C9 82 D0 E9 85 E6 48 45 E4 85 E4 68 29 7F A0 00     02,D0
  1E90  99 4F D1 20 E2 1E 08 C9 03 F0 0B 28 90 CF A9 01
  1EA0  D0 12 A9 02 D0 0E 28 20 EB FF C5 E4 F0 04 A9 03
  1EB0  D0 02 A9 00 85 E5 A9 00 8D 03 02 A5 E7 A4 E8 85
  1EC0  E2 84 E3 A9 F9 A0 1E 8D 18 02 8C 19 02 A5 79 A4
  1ED0  7A C8 85 E7 84 E8 A9 00 85 E4 60 A0 00 B1 E7 4C
  1EE0  E6 1E A0 00 91 E7 A8 E6 E7 D0 02 E6 E8 A5 E8 C5
  1EF0  E3 90 04 A5 E7 C5 E2 98 60 20 DB 1E B0 01 60 20
  1F00  6C A8 A5 E5 F0 1D C9 01 D0 07 A9 37 A0 1F 4C 20
  1F10  1F C9 03 D0 07 A9 47 A0 1F 4C 20 1F A9 99 A0 A1
  1F20  20 C3 A8 A9 92 A0 A1 20 C3 A8 A9 BA A0 FF 8D 18  Cegmon 46,FB
  1F30  02 8C 19 02 A9 0D 60 0A 0D 4F 55 54 20 4F 46 20
  1F40  4D 45 4D 4F 52 59 00 0A 0D 43 48 45 43 4B 53 55
  1F50  4D 20 45 52 52 4F 52 00 A9 84 A0 1F 8D 1A 02 8C
  1F60  1B 02 A9 94 A0 1F 85 04 84 05 A9 99 A0 00 85 13
  1F70  84 14 A9 82 85 E4 20 B1 FC A2 12 A0 00 86 C3 84
  1F80  C4 4C F6 A5 C9 0A F0 0B 09 80 48 45 E4 85 E4 68
  1F90  20 B1 FC 60 48 98 48 A9 C3 A0 A8 85 04 84 05 A9
  1FA0  69 A0 FF 8D 1A 02 8C 1B 02 A9 0D 20 B4 1F A9 03  Cegmon 9B
  1FB0  20 B4 1F A5 E4 20 B1 FC 68 A8 68 4C C3 A8
```

The baud rate chosen for BASICODE is 1200 baud. The modification to the C1P wiring to produce this baud rate has already been published twice; in KAOS July 82/P3 and in KAOS Sept. 83/P2. KAOS will supply a copy of either page if you forward a SAE. In my opinion, 300 baud would have been a more reliable baud rate. While 1200 works fine most times when saving/loading to/from the same cassette recorder, it is much less reliable when you try to load from a cassette recorded on another machine.

Next month, there will be an article on BASICODE specifications, for those in KAOS who do not own C1Ps, plus the BASICODE protocol, for the writing of Basic programs for use on other types of machines. The protocol is not so simple as it would at first appear.

— SUPERBOARD —

E.B.C. UPDATE

Mark Howell has done some extensive modifications to adapt the Extended Basic Commands program to work in perfect harmony with DABUG III in either the 24 x 24 or 12 x 48 modes of his series 2 Superboard. He has also done some changes to improve the code, preventing wrap-around in the renumber routine, and detecting if an existing line will be overwritten by the copy function.

So if any previous purchaser of E.B.C. would like a Dabug version, please send your E.B.C. tape, together with a 45¢ stamp, to OSUG, and I will return it with the modified program recorded, plus written details of the operational changes, and the new relocation data.

- -

RESMON III

Mark Howell has also extensively modified the Dabug III monitor to give it some excellent features. His RESMON III offers:-

A new keyboard algorithm. (see KAOS 3/9 and 3/8)
The complete character set can be printed in 12 x 48 mode.
A new error code correction routine. (see On Error Goto, MICRO #51)
CTRL W resets stack before exiting to $0000. No warmstart OM ERROR.
CTRL E resets page 2 vectors and exits in 24 x 24 mode to M/C Monitor.
CTRL P resets ACIA to full speed and sets printer flag.
CTRL T resets ACIA to tape speed and clears printer flag.
CTRL V calls a tape view routine, using the space bar to exit.
The output routine at $FF69 does a printer flag check.
Two other USR jumps are available on BREAK.

For more details, write to Mark at

- -

THE LAST GREAT O.S.U.G. PROGRAMMING COMPETITION

By the time you read this, the voting will be coming in for the final OSUG programming competition that I will run.
If you are an entrant, then this is a reminder that there is not much more time to register your votes. April 2nd is the deadline!

The competition was a fair success, with 11 entrants and a total of 14 entries in both sections of the competition. I was a little disappointed that there were only 5 entries in the 32 x 64 section, but it all turned out well, as all entries fitted on a C60 cassette.

Most of the entries were games. There were 7 arcade games, 2 educational games, 2 educational programs , 2 utilities, and an adventure. There were no major problems with running any of the programs.

Next month, I will list the results of the Competition.

- -

NEXT MONTH'S

Apologies for the missing article on the Word Game Solver. I will present it next month. More to come on NOS BASICODE. Anyone considering the purchase of an Apple (or a clone) should wait for next month's KAOS; it could save one a great deal of cash.

COMING

Bernie Wills' high-res driver program. A neat way to set up a Teletype. All those other things I mentioned last month, and keep putting off!

Ed Richardson.

THE BEGINNING MACHINE LANGUAGE PROGRAMMER.....Part The Last

by David Dodds

Most microprocessors provide facilities which enable external devices to signal the CPU when they need attention. The CPU looks for these signals during each instruction cycle. If a signal is present the CPU responds by interrupting the current program and processing a special program designed to service the devices providing the signal. This facility is known as an Interrupt system.

The 6502 has two types of hardware interrupt which form a priority hierarchy. These are known as:

NMI (Non Maskable Interrupt)
and IRQ (Interrupt ReQuest)

NMI is used by very high priority devices (Those which require immediate attention). Typically the NMI would be used by power loss detection circuits to signal impending power failure. The CPU will always respond to an NMI request.

IRQ is the more commonly used of the two. The reaction of the CPU to an IRQ depends on the state of the Interrupt Disable bit (I) in the status register. The Interrupt disable is set (true or one) if the CPU has been instructed not to respond to an IRQ. Providing the Interrupt Disable is clear the CPU will respond to the IRQ.

The CPU checks for the presence of an interrupt signal before commencing each new machine instruction. When responding to an interrupt the CPU will

1 push the programme counter onto the stack
2 push the status register onto the stack
3 set the Interrupt Disable bit (disable interrupts)
4 reset the programme counter to the address of the appropriate interrupt service routine.

The 6502 expects that the address of the interrupt service routines will be stored at specific locations in memory. These locations are known as the Interrupt vectors.
Each of the interrupt types has its own vector as follows:

NMI $FFFA/B
IRQ $FFFE/F

The address contained in whichever location is appropriate to the interrupt is what is loaded into the program counter. The CPU then continues to process the interrupt service routine until instucted to return to what it was doing before the interrupt. The instruction which does this is known as Return from Interrupt or RTI in mnemonic form.

An RTI instruction causes the CPU to
pull the status register from the stack
pull the program counter from the stack.
The program counter is not incremented as it would be in an RTS. This is because the instruction now pointed to by the program counter had not been processed when the interrupt occurred. Since an RTI resets the entire Status register every bit may be affected.

In addition to the hardware interrupts the 6502 also provides for a software interrupt which is known mnemonically as BRK (Break).

BRK shares the $FFFE/F vector with IRQ. Before saving the status register and the programme counter the BRK instruction causes two things to occur. Firstly the Break bit in the status register is set to 1 indicating that BRK has occurred. Secondly the programme counter is incremented by two. Note this carefully. An RTI from a BRK does not return programme execution to the point it left from. Special arrangements need to be made for this to occur.

The programmer can detect whether an IRQ or BRK has occurred by examining the Break flag (B) which is bit 4 of the status register. The usual technique for this is

```
        PLA             ;pull status register
        PHA             ;restore stack pointer
        AND #%00010000  ;mask break flag
        BNE DO.BRK      ;process BRK routine.
```

BRK is most commonly used as a program debugging tool but can be used as a means of transferring programme control back to a disk operating system or monitor. As a debugging aid BRK would be used to replace the first byte of an instruction at a selected point in a program. When BRK is encountered it causes a jump to a routine which displays register contents and allows memory to be examined. Possibly the simplest way to implement a register examine facility would be to store the A,X and Y registers in sequential memory locations along with the status register which was obtained from the stack. The OSI monitor could then be used to step through the locations to view the contents of the registers.

Well thats it! I would have liked to have gone into more detail but for obvious reasons this just wasn't practical. (I have no intentions of writing a monthly article for another 2 years). For those of you who want to know more I would recommend the 6502 programming books written by L. Levanthal and published by Osborne/McGraw-Hill. At around $20 each they are not cheap but well worth the money if you want to get serious about machine language.

Don't forget; Next month these pages just might be blank:- it's up to you!

THE MEETING WAS KAOS

by King Corky

Last month we had our annual Trash & Treasure sale and it went off reasonably well. George and the boys from Compsoft seemed to have been the biggest hit with the Rabble boards and floppy drives. I made $10 and spent $10, going home with as much as I took, but of a different variety, but that's what it's all about, isn't it?

As far as the meeting side of things went, we were informed that Michael Lemaire has finally admitted defeat. He has handed over the, (almost finished), front-end for the TAB to our chairman, Warren for final clean-up and testing. Michael really has a lot on his plate at Compsoft and the time he had to spend on the front-end was costing the shop heaps of money. So Warren now has the task of completing this much awaited software that will allow all of us who are fortunate enough to have a modem, to access the TAB and use their upload and download facilities for both the public-domain software and the OSI/Rabble software that we hoped will be placed on the system.

The hard-copy of Adam Dickson's Floating-point Forth is available through the club secretary, Rosemary, free.

David Anear had a chance to do a quick evaluation of the new Apple arrival, McIntosh. This is a compact, portable system that has practically everything on it that anyone could need. It runs a 68000, has a not-too-small for-a-portable 10" screen, 3.5" 400K floppies, and a complete interfacing set-up that will allow you to drive almost any external device that is currently available. Most inputing can be done with the supplied Mouse, it comes with two software packages, a word processor and a graphics pack, all this for around $3000. Half of Microsoft's resources are now devoted to writing software for the McIntosh, so we should see some impressive packages coming out soon. All in all, a very good buy, (wish I was getting paid for all these ads).

From Paul Dodd at Compsoft, the CP/M and Multipurpose floppy controller will be available very soon, (please hurry), at around $390 the pair, including Z80 CPU, 64K RAM, and with the advantage of being able to read both 8" and 5.25" floppies, single or double density, without any hardware switching required. The Demo software pack, a terminal routine, will be free with the boards.

Now for the important news. By the time you read this, our beloved spare parts/system/advice supplier, George Nikolaidis, will be well-and-truly married and may even be back from his honeymoon, (he rented the rear room at the factory, cheapskate)!! Well done George and many happy returns? No, that's not right, is it? Best wishes, have a loooong and happy marriage and give Irene a big kiss from each of us, (he won't let us do it personally). Congratulations to you both.

Most of us should know by now that the Oz/Rabble boys are on their way to the USA to show off their favourite daughter, the Rabble 65, we wish them the best of luck and hope they bring back lots of multi-million dollar orders for what must be one of the best value personal computing systems in the world, at least we think so.

Finally, if anyone out there would like a chance to chat to another OSI user via their computer, or exchange some software they have written, (this is through a modem of course), I am available most nights except Monday and Friday on 03 763 6497. This applies to both Dino and Bill R. as well. The COMMS smart terminal is a ripper for this sort of thing, with indirect file use while still running the terminal routine. See you soon. Bye.

OSI HISTORY Part 4

by Eric Lindsay

Comp-Dos 1.2 links to OS65D3.0, 3.1 or 3.2 to add a PUT filename command that automatically places the filename in the directory when you PUT a file to disc. It adds a DR command, which prints the directory, and can be called from BASIC or from the operating system. Other additional commands available from the operating system or from BASIC include DE, which deletes a file name from the directory (but does not remove it from the disc, thus giving you a chance to recover it if you make a mistake. RN lets you rename a file. The existing sector directory command, which previously indicated the number of sectors on a track, will now provide details of any specified range of tracks (the OSI system lets you produce tracks with almost any number of sectors, with various lengths per sector). A new command, CO, loads the existing OSI copier utility in place of the language. CL provides a machine code screen clear, not previously available from BASIC.

The CR command creates a new entry in the directory, and you can specify either how long a file you want, and let the operating system decide where it will fit, or else you can specify the start and end track, if you need it to occupy specific tracks. Since this does not initialize the tracks to be used, there is also a ZE command which can be used to initialize the contents of data files to zero. This command is not needed for saving programs from BASIC.

There is also a new command to simplify the use of random access files, by letting you create space for a disc buffer from within BASIC. The previous system demanded that you run a utility program when powering up the system. You can also release this buffer space from BASIC.

Three new errormesages are added, indicating that a file name already exists, that the disc is full, and that a file you are attempting to create will overrun the track range of an existing file. Comp-Dos 1.2 comes with a 17 page manual explaining clearly how to use the new commands. The DOS extensions were written by Compsoft's owner George Nikolaidis.

OSI came out with a very much revised version of their OS65D. This is version 3.3, which comes with a collection of about 250 pages of notes and tutorial, and several tutorial discs. Most of the changes relate to how BASIC is handled, and the video display now emulates a Hazeltine 1420 intelligent terminal. This means that you have various (ESC) commands to run the cursor round the screen (making the screen editor in Comp-Dos 1.2 superfluous), plus editing features for BASIC (not as good, in my opinion, as those provided in David Anear's Dabug editor). You can also set windows on the screen, select colours (if you run a colour monitor), and BASIC now provides PRINT AT and PRINT USING facilities. Both upper and lower case can be used for all BASIC commands. You can alter the cursor to any character, and the input prompt to any character. A disk find command lets you search any sequential file for a specified string, thus easing the writing of some business programs.

Paul G. Dodd and Michael J. Lemaire of the KAOS group set out to adapt Comp-Dos 1.2 to suit OS65D3.3. The result is Comp-Dos 1.3, of which I have only used the preliminary version. Only one extra command is added, and this simply lets you change the DOS prompt to any specified string. Comp-Dos 1.3 gives its own name as the prompt, unlike the OSI version, which gives A or B, depending upon which disc drive is currently selected.

File names can now contain 14 letters, an improvement over the 6 letter filenames previously allowed. I would rather like to see the addition of disc names, since this change is also well documented by various user groups, but this may well appear in future upgrades. The directory now contains a neater heading, and is listed in two columns. Selection of which drive is now implicit, by adding %A or %B after any command in which the drive number is relevant. This saves a significant amount of time. Filenames can now contain a question mark as a wildcard character, so that you do not have to type the entire filename when entering commands. The delete command can be followed by a *. When this is used, all file entries are deleted. To ensure this is desired, the computer first generates a "Delete all file ?" message.

Error messages in OS65D normally consist of cryptic messages like "Error 1". Comp-Dos 1.3 gives the error number, and prints the error message in full.

While I can't say that I am entirely enchanted with OS65D3.3, since it seems somewhat slower than the earlier versions, I must admit that Comp-Dos 1.3 makes it much more pleasant to work with. The upgrade cost $15, but contact Compsoft for the full price if you are not upgrading from a previous version. Source code is not available for Comp-Dos 1.3, however you can purchase the source, on disc, for Comp-Dos 1.2 for about $15. Both versions are excellent Australian products, at very reasonable prices, and both make using the OSI much easier.

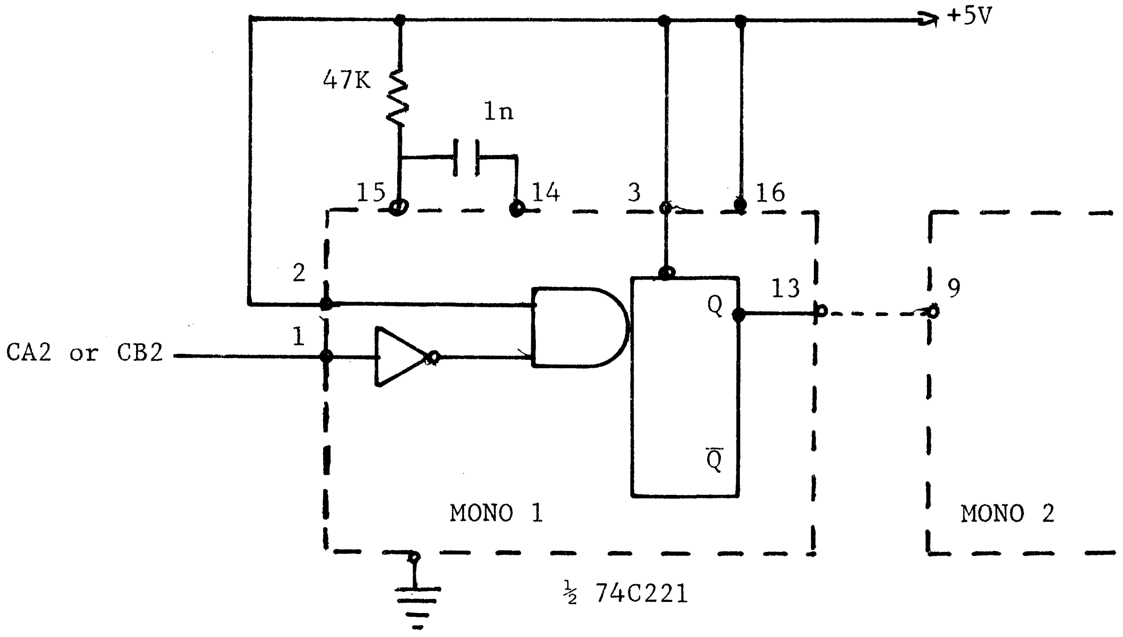
Continued next month

MODIFICATIONS TO MICROBEE EPROM PROGRAMMER FOR USE WITH 6522 VIA OR 6521 PIA

by Nino Benci

The Microbee Eprom programmer was chosen by me because of its versatility in that it has a personality socket. This enables the programming of five different types of Eproms.

The programmer is also easier to use in that it only needs an eight bit port and one handshake line, this is supplied by either the CA2 or CB2 handshake lines. The only modification needed is to the wiring of monstable 1. This is shown below.



Once this is done, all that is needed is to program the VIA so that it supplies the handshake signal. Using port A of the VIA, a small routine is supplied below.

```
OUTPUT ROUTINE                INPUT ROUTINE
LDA #$FF                      LDA 5$00
STA $C033                     STA $C033
LDA #$10                      LDA #$10
STA $C03C                     STA $C03C
```

The VIA will send a strobe out whether it is an output port or an input port. The data line resistors may have to be changed, ie. decreased to about 220ohm. For further information, refer to the article in ETI magazine.

ITEMS FOR SALE.

Collected by John Whitehead.

The list below is in the order I received them, and uses the following abbreviations:- (I) = Item. (D) = Description. (K) = Full details in KAOS. (W) = Works on. (R) = Minimum RAM needed. (E) = Extra required for item to work. (P) = Price. (A) = Available from.
When ordering any items, please send full details of your computer.

....

(I) Replacement Character Generators
(D) All alphanumerics in OSI set are a pixel row shorter in height, giving improved readability. ENHANCED set: Chess pieces, trains, more cars & planes, other improved graphics. SCIENTIFIC set: Alfa as above, plus Greek letters, mathematical & scientific symbols. MEDIUM RESOLUTION: 256 block characters giving a resolution of 48Hor x 96Vert on 24 x 24, or 128 x 128 on 64 x 32 screen.
(K) Aug 82. Sep 82. Aug 83. Oct 83
(W) C1. Superboard. Several sold to C4 users.
(E) Toggle switch for hardware selection or 74LS174 or 74LS74 for software selection. Details included.
(P) Any one of above in a 2716 $10.50 inc p&p. Any two of above in a 2732 $14.00 inc p&p. 10% discount for two or more EPROMs.
Medium res software for C1 keyboard machines with Cegmon, Dabug or Synmon on cassette $2.50 ($1 p&p if ordered alone). Software is free if you send a cassette (without a case).
(A) B. Wills,

....

(I) EPROM Extender.
(D) Printed circuit board to add 4 or 8k of ROM.
(K) Aug 82.
(W) C1P superboard or C4P
(E) Any monitor rom in a 2716 or 2732. e.g. Dabug.
(P) $8.50 + $1 p&p
(A) R. A. Patterson, 20 Donegal Street, Salisbury Downs, S.A. 5108

(I) 16k CMOS RAM Board
(D) Adds 16k of 6116 RAM. to your computer.
(W) C1P superboard or C4P.
(P) PCB $20.00 or $25.00 drilled + $2 p&p.
(A) R. A. Patterson.

(I) 40 Pin Socket Expansion
(D) Four 40 pin sockets on a printed circuit board.
(K) Dec 83.
(W) C1P or modified C1P-C4P with j1 socket.
(P) $15.00 or $20.00 drilled + $2.00 p&p.
(A) R. A. Patterson.

....

(I) Forth Interpreter in BASIC.
(D) Allows you to use Forth on your system (does not require disk). Ideal way to learn to program in this popular language. Includes major Forth

features such as IF...THEN...ELSE, DO...LOOP, memory put and fetch, stack manipulation and floating point arithmetic. Full documentation supplied with cassette.
(W) Any Ohio and virtually any BASIC computer.
(R) 8k.
(P) $12.50 (includes certified postage).
(A) Stuart Thomas,

....

(I) Compdos Relocator.
(D) A verson of BEXEC* that relocates COMPDOS 1.2 if memory size is different from (say) 40k. Saves 2 tracks/disk while retaining ability to boot COMPDOS on different configurations. Optionally includes revised keyboard algorithm as in KAOS 3.8 & 3.9.
(W) C1P Superboard etc with 5.25 SSSD disk drive.
(R) 16k. The more the better.
(E) 5.25 single density disk drive and a COMPDOS licence.
(P) $4 on your disk at KAOS meetings or $15 inc p&p. $2.00 + $0.50 p&p for printed listing.
(A) Rodney Eisfelder,

(I) LISTP.
(D) Transfers PICO-DOS programs to OS65D-V3 or COMPDOS 1.2
(W) C1P Superboard etc.
(R) 28k. Could be made for 24k.
(E) 5.25 single density disk drive.
(P) As above.
(A) Rodney Eisfelder.

(I) Banner.
(D) Makes banners of any length using large uppercase, space, fullstop & asterisk char. writing on screen or printer. Can be easily extended.
(R) 16k.
(E) Printer and OS65D-V3 or COMPDOS. Program is set up for Microline 80.
(P) As above.
(A) Rodney Eisfelder.

(I) REPAY and REPTAB.
(D) REPAY calculates a schedule of morgage repayments for a given loan, period & interest. REPTAB prints a table showing size of instalments. Algorithm based on Texus Inst. SR56 applications library.
(W) C1P Superboard etc.
(R) 4k for cassette, 16k for disk.
(E) Printer essential for REPTAB
(P) As above.
(A) Rodney Eisfelder.

(I) Faster Disk BASIC.
(D) Runs scientific programs upto 40% faster and most programs run 5 to 10% faster. Includes CALL statement for calling M/Code and large random access file bug fix.
(K) Vol 2, #8,9,11 & 12.
(W) C1P superboard etc.
(R) 16k. The more the better.
(P) As above.
(A) Rodney Eisfelder.

(I) UTE
(D) An all-in-one utility program. Replaces OS65D-V3 utilities. CREATE, DIR, SECDIR, RENAME, DELETE, TRACE, ZERO, Decimal to Hex, Copy by file name, Help menu, Change Drives etc.
(W) C1P Superboard etc.
(R) 32k. Can be modified for 24k.
(E) 5.25 SSSD disk drive.
(P) As above.
(A) Rodney Eisfelder.

(I) Disassembled Listings of BASIC.
(D) Disassembled and partly commented printed listings of ROM or DISK BASIC. ROM version has been sold by LOOKY VIDEO and if you have bought this since July 1981, please let me know.
(P) $10 + $2 p&p.
(A) Rodney Eisfelder.

....

(I) EXMON in EPROM
(D) Improved BASIC compatible Extended monitor containing 26 routines for machine code work including disasembly.
(K) Sep 83 and Nov 82.
(W) C1P Superboard.
(E) Any expansion board that will take a 2716 EPROM at $E000 and a copy of original EXMON manual.
(P) $10 + $1 p&p.
(A) John Whitehead,

(I) EXMON PLUS.
(D) Extra EXMON type routines. Hex don't care search, Compare blocks of memory, Disassemble to a given address for a printer, Relocate a BASIC program to run at any hex address, Memory size non destructive test, Uncrash a BASIC program, Display memory in hex and ASCII at the same time. Also Real Time Clock and Teletype routine that needs RAM at $CF00 to $CFFF.
(K) Nov 83.
(W) C1P Superboard.
(E) My EXMON at $E800. and original EXMON manual.
(P) 2716 EPROM to work at $E000. $10 + $1 p&p. or 2732 EPROM that also includes EXMON to work at $E000. $12 + $1 p&p.
(A) John Whitehead

(I) BASIC 3 with Garbage Bug Fixed.
(D) Replace BASIC 3 ROM with a 2716 EPROM to completly fix the bug. (Not the one or two byte incomplete fix)
(K) July 82.
(P) $10 + $1 p&p.
(A) John Whitehead.

(I) WP6502 V1.2 in EPROM
(D) WP6502 altered for DABUG in 48 x 12 at $8000 to $8FFF and gives over 7k of workspace on a 8k superboard. Can include a Model 15, 33 or 35 teletype routine.
(W) Superboard series II with Dabug.
(E) Original WP6502 V1.2 tape and manual plus expansion for a 2732 EPROM.
(P) $12 + $1 p&p.

(I) DABUG 3J.
(D) Correctly decoded keyboard is its main advantage.
(K) Oct 83.
(W) C1P Superboard.
(P) $12 + $1 p&p.
(A) John Whitehead.

FOR SALE

Superboard Series 2, in case with power supply. Has BASIC 4 and parallel port. Lots of software.
$200 Harry Moores,

Rabble Ozi expansion board, 32K CMOS RAM (6116), FDC, RTC, VIA and PIA outputs, ROM expansion, 2 dual programmable sound generators.
$200 or $120 without CMOS RAMs ono   Noel Coward

OSI C2-4P with 32K user RAM, 4K video RAM, 24K EPROM, inbuilt DAC, sound (tones), speech, serial (RS232), parallel, joystick and cassette ports. Cassette and printer ports crystal clocked for 300, 600,1200 and 2400 baud. EPROM has extended BASIC, Extended Monitor (with additions), Assembler /Editor, WP6502, Dabug3, 2732 EPROM burner program, text to speech routine, and simple monitor. Additional BASIC functions include: PRINTAT statement, INKY (inkey) statement (with rewritten keyboard routine to fetch characters or ASCII values), DEL for block line deletion, line renumbering, program packing, OSUG BASIC4, fixed garbage bug routine, set and reset graphics points, talk statement similar in syntax to PRINT statement, etc. Power supply upgraded to 240V input/5V at 10 Amp +/- 12V at 1 Amp output, 240V cooling fan. 540 video board upgraded for 512 characters in EPROM (approx. 900 different with reverse), 128x128 graphics resolution, reverse characters, flashing characters. Can have graphics and Alpha characters on screen at same time, and characters can be reversed and flashing at same time. ALL ABOVE BUILT INTO ORIGINAL CASE.

Completely rewritten manual with all functions OSI forgot and additions included, redrafted circuit diagrams covering all mods (drawn on drafting board, not just sketches). Also complete memory map showing allocation/usage /function of all memory locations in above machine.

Also EPROM burner for 2732's on 2"x3" card and OSI Chess etc in EPROM's on 2"x3" card (plug into parallel port), Forth plus over 30 utilities and games on cassette.
$700. Wayne Geary,

WANTED   8K EPROM Card to suit Tasker buss system. Must be in reasonable condition. Price negotiable.
Ph   and ask for Mark.

Registered by Australia Post
Publication No. VBG4212

If undeliverable return to
KAOS, 10 Forbes St
Essendon, Victoria 3040